Titre: Title:	User trajectory prediction in mobile wireless networks using quantum reservoir computing
Auteurs: Authors:	Zoubeir Mlika, Soumaya Cherkaoui, Jean Frederic Laprade, & Simon Corbeil-Letourneau
Date:	2023
Type:	Article de revue / Article
Référence: Citation:	Mlika, Z., Cherkaoui, S., Laprade, J. F., & Corbeil-Letourneau, S. (2023). User trajectory prediction in mobile wireless networks using quantum reservoir computing. IET Quantum Communication, 4(3), 125-135. https://doi.org/10.1049/qtc2.12061

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/54832/
Version:	Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	CC BY-NC-ND

Titre de la revue: Journal Title:	IET Quantum Communication (vol. 4, no. 3)
Maison d'édition: Publisher:	John Wiley and Sons Inc
URL officiel: Official URL:	https://doi.org/10.1049/qtc2.12061
Mention légale: Legal notice:	This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made. © 2023 The Authors. IET Quantum Communication published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

Received: 9 January 2023 | Revised: 29 March 2023 | Accepted: 26 May 2023

DOI: 10.1049/qtc2.12061

IET Quantum Communication

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

User trajectory prediction in mobile wireless networks using quantum reservoir computing

Zoubeir Mlika[1] | Soumaya Cherkaoui[2] | Jean Frédéric Laprade[3] | Simon Corbeil-Letourneau[4]

[1]Department of Electrical Engineering, Polytechnique Montreal, Montréal, Québec, Canada

[2]Department of Computer and Software Engineering, Polytechnique Montreal, Montréal, Québec, Canada

[3]Institut quantique, Université de Sherbrooke, Sherbrooke, Québec, Canada

[4]Thales Digital Solutions, Saint-Laurent, Québec, Canada

Correspondence

Zoubeir Mlika.
Email: mlika.zoubeir@gmail.com

Funding information

Natural Sciences and Engineering Research Council of Canada; Fonds de recherche du Québec - Nature et technologies

Abstract

This paper applies a quantum machine learning technique to predict mobile users' trajectories in mobile wireless networks by using an approach called quantum reservoir computing (QRC). Mobile users' trajectories prediction belongs to the task of temporal information processing, and it is a mobility management problem that is essential for self-organising and autonomous 6G networks. Our aim is to accurately predict the future positions of mobile users in wireless networks using QRC. To do so, the authors use a real-world time series dataset to model mobile users' trajectories. The QRC approach has two components: reservoir computing (RC) and quantum computing (QC). In RC, the training is more computational-efficient than the training of simple recurrent neural networks since, in RC, only the weights of the output layer are trainable. The internal part of RC is what is called the reservoir. For the RC to perform well, the weights of the reservoir should be chosen carefully to create highly complex and non-linear dynamics. The QC is used to create such dynamical reservoir that maps the input time series into higher dimensional computational space composed of dynamical states. After obtaining the high-dimensional dynamical states, a simple linear regression is performed to train the output weights and, thus, the prediction of the mobile users' trajectories can be performed efficiently. In this study, we apply a QRC approach based on the Hamiltonian time evolution of a quantum system. The authors simulate the time evolution using IBM gate-based quantum computers, and they show in the experimental results that the use of QRC to predict the mobile users' trajectories with only a few qubits is efficient and can outperform the classical approaches such as the long short-term memory approach and the echo-state networks approach.

KEYWORDS

learning (artificial intelligence), quantum computing

1 | INTRODUCTION

One of the fundamental theories in physics is quantum mechanics, which is the foundation of all quantum physics such as quantum chemistry, quantum technology and quantum information science. Quantum mechanics describes the physical properties of nature at the scale of atoms and subatomic particles [1]. The phenomena of quantum mechanics such as interference, superposition and entanglement can be exploited to create a new paradigm of computation known as quantum computing [2, 3]. Quantum computers are devices that perform quantum computations [4] and are shown to have a great potential of fast information processing. For example, the mathematical problem of integer factorisation is shown to be solved efficiently on a quantum computer despite the fact that it is believed to be intractable on a classical computer [5].

In this study, we propose to use quantum mechanics to solve the machine learning task of mobility prediction in mobile wireless networks. This task is nonlinear and belongs to real-world temporal information processing tasks such as time-dependent signal processing, stock-market prediction, natural language processing etc. The combination of quantum mechanics and machine learning can help solving these real-world temporal information processing tasks. The proposed machine learning approach belongs to the reservoir computing (RC) framework, which is inspired by how the brain processes information. Unlike complex recurrent neural networks (RNN) where training is a complex procedure, the RC framework uses the so-called reservoir to project the input signals to a higher dimensional space, thus producing a highly dynamical network capable of emulating nonlinear and temporal information processing systems. The reservoir is composed of hidden nodes and visible nodes. The combination of both produces a high-dimensional signal. The training is performed only at the output layer using simple regression analysis [6–8]. To perfectly emulate non-linear dynamical systems, the dynamics of the reservoir must involve adequate nonlinearity and memory [9]. We apply a recently proposed quantum RC (QRC) framework [10] to the prediction of user trajectory in mobile networks. Mobility management is an important and challenging problem in future 5G/6G networks. In fact, future communication networks will need to provide advanced services for various application areas including smart cities and epidemic prevention [11, 12] with diverse requirements in terms of ultra-low latency, data rates and massive connectivity. For critical communications and service delivery in particular, issues such as disconnections or service level agreement (SLA) violations can be problematic. A reactive mobility management approach fails to perform well in highly dynamic networks [13, 14]. Therefore, a proactive [14] mobility management algorithm is required where the mobility patterns of the users can be predicted in advance.

1.1 | Related works

The prediction of user trajectory prediction in wireless networks is not a new topic and has been studied previously in different works [15–20]. However, to the best of our knowledge, no previous work applied quantum mechanics and machine learning to solve the challenging problem of trajectory prediction in mobile wireless networks. Quantum mechanics and machine learning have been proposed to solve machine learning tasks such as classification and prediction [10, 21–27], but they were not applied to a practical mobile wireless network problem.

On the one hand, in Ref. [15], the authors proposed a mobility prediction approach based on RNN. Precisely, the authors proposed to use a gated recurrent unit (GRU) and a long short-term memory (LSTM) technique to predict the global positioning system (GPS) coordinates of a mobile user; taken from the Geolife dataset [28–30]. Their approaches were initialised with a pre-processing step to reduce the number of points and thus the prediction complexity in the time series data. In Ref. [16], the same authors proposed the same mobility prediction approaches but using bidirectional analysis and three different datasets: Geolife, open street map and T-drive trajectory. In Ref. [17], the authors studied two data-driven mobility prediction algorithms in vehicular networks and measured how inaccurate their prediction accuracy was. To overcome the imperfect mobility prediction problem, the authors designed a new scheduling system to offload cellular traffic to vehicular network and showed that prediction accuracy was improved. In Ref. [18], the authors proposed a decentralised traffic management scheme to predict vehicle trajectory. Each vehicle predicted its own trajectory by training an LSTM model. The proposed approach used gossip learning and iterative model averaging is used to build a global model. The evaluation was based on urban vehicular network environment where the dataset of the Luxembourg SUMO Traffic scenario [31] was used. In Ref. [19], the authors proposed an RNN-based LSTM approach to predict vehicle trajectory using a taxi dataset of 442 taxis running in Porto, Portugal. The results showed the effectiveness of the prediction with a prediction performance higher than 89%. The improvement was observed especially when more data are available prior to the next prediction (prediction of the next cell). In Ref. [20], the authors studied the problem of power control and trajectory planning in unmanned aerial vehicle networks. The aim was to maximise the sum rate while guaranteeing the rate requirement of the users. First, an UAV placement algorithm was proposed based on Q-learning to position the UAVs. Then, using the real data of the users collected from Twitter, the authors proposed an echo-state network (ESN) to predict the future position of the users. Finally, a Q-learning algorithm was proposed to predict the UAV positions in future timesteps. The results showed that as the size of the reservoir in ESN increases, the prediction increases, and a sum rate gain of 17% was obtained.

On the other hand, in Ref. [21], the authors proposed a QRC system where the dynamics of the reservoir evolve according to Hamiltonian evolution in the fully connected transverse field Ising (FC-TFI) model. The authors studied the memory capacity and the accuracy of the proposed QRC by varying different parameters such as the inter-spin interactions of the FC-TFI model and the time evolution scale. They showed the existence of an optimal time evolution scale at which the capacity of the QRC is maximised. Finally, they applied their method on stock prediction data. In Ref. [23], the authors aimed to overcome the random walk dilemma for financial time series prediction using a quantum-inspired hybrid method. Their method used a qubit multilayer perceptron (QuMLP) and a quantum-inspired evolutionary algorithm (QIEA). The QIEA was a search-based algorithm that trains the QuMLP to determine parameters like the maximum number of time lags to represent the financial time series, the number of units in the QuMLP hidden layer. Four time series datasets were evaluated that correspond to the daily records of Nasdaq stock market, and the results were shown to be superior compared to classical MLP. In Ref. [25], the authors proposed to use superconducting quantum computing devices as the reservoir in a QRC system. They showed that the inherent noise characterising nowadays quantum computers is advantageous and helped in providing dissipative dynamics capable to learn a dynamical system and solve a temporal information processing task. The authors studied the prediction of the non-linear autoregressive moving average (NARMA) time series as well as an experimental-based classification problem to classify three objects: one LEGO cube, one polylactic acid (PLA) cube and PLA sphere. The time series for the classification problem were generated by the triboelectric nanogenerator (TENG) sensor of a robotic gripper grabbing the objects. In both the prediction and the classification problems, the proposed gate based QRC approach showed a higher performance than classical linear regression or classification models. The authors concluded that a noisy quantum device can potentially work as a reservoir computer, and notably, that the undesirable quantum noise can be used as a rich computation resource for machine learning tasks. In Ref. [26], the authors proposed a hybrid quantum-classical network to classify non-trivial datasets such as finance and MNIST data. They discussed how their approach can have overfitting problems as in the classical case and they propose different regularisation techniques to solve the problem. In Ref. [27], the authors discussed the use of the Harrow Hassidim Lloyd (HHL) algorithm in the regression problem. They designed a 7 − qubit quantum circuit to solve a three-variable regression problem, using only elementary quantum gates. They implemented the stochastic group leaders optimisation algorithm (GLOA) and discussed on the advantages of using GLOA to create low-cost circuits for simulating the Hamiltonian. In Ref. [10], the authors proposed to exploit the natural quantum dynamics of ensemble systems to solve a temporal information processing task using QRC. The proposed QRC framework was shown to enable ensemble quantum systems to universally emulate non-linear dynamical systems including classical chaos. The authors performed several numerical experiments for quantum systems with 5−7 qubits and showed their superiority in terms of computational capabilities compared to conventional RNNs composed of 100−500 nodes.

1.2 | Contributions

- We applied the QRC approach to predict mobile trajectories in wireless communications.
- We used the fully connected transverse field Ising (FC-TFI) to implement the QRC approach.
- We implemented the QRC approach by estimating the time evolution of the Hamiltonian using IBM gate-based quantum simulators and computers.
- We compared the QRC approach to the classical benchmark approaches, LSTM and ESN.

1.3 | Paper organisation

The rest of the paper is organised as follows: Section 2 first introduces the basic concepts of quantum mechanics and then describes the QRC framework used to predict user trajectory. Section 3 describes classical state-of-the-art solutions. Section 4 describes the real-world experimental setups and illustrates important results and conclusions. Finally, Section 5 draws some important conclusions.

2 | QUANTUM RESERVOIR COMPUTING APPROACH

The idea of the proposed QRC framework is essentially based on applying the FC-TFI model introduced in Ref. [10, 21]. In this framework, the dynamics of the recurrent part of the network (the reservoir) are governed by the dynamics of rich quantum mechanical systems. Before describing in depth, the QRC framework, we start by introducing some basic concepts about quantum computation and quantum information.

2.1 | Qubit state

Contrary to the classical bit—the minimum unit of information in classical computing, the quantum bit (or simply qubit) is the minimum unit of quantum information in quantum computing. A qubit is a two-state quantum-mechanical system such as the spin of the electron with the two states of spin-up and spin-down. As opposed to a classical system in which a bit is only in one state at a time, a qubit, based on the laws of quantum mechanics, can be simultaneously in a coherent superposition of the two states. Mathematically, a qubit is a vector in a two-dimensional complex vector space spanned by the orthonormal basis $\left\{\overrightarrow{0}, \overrightarrow{1}\right\}$. In quantum mechanics, the vectors $\overrightarrow{0}$ and $\overrightarrow{1}$ of the basis $\left\{\overrightarrow{0}, \overrightarrow{1}\right\}$ are conventionally written in the Dirac notation (or the bra-ket notation) as $|0\rangle$ and $|1\rangle$, respectively. A pure qubit state can be in any coherent superposition (linear combinations) of the two basis states $|0\rangle$ and $|1\rangle$, that is, a qubit can be written as [2]

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad (1)$$

where α and β are complex numbers that are called the probability amplitudes. When a qubit is measured, analogously, when a bit is read, the outcome of the measurement is state $|0\rangle$ with probability $|\alpha|^2$, and state $|1\rangle$ with probability $|\beta|^2$. Since the probabilities must sum to 1, the coefficients obey to $|\alpha|^2 + |\beta|^2 = 1$. In other words, the qubit's state must be normalised to length 1. In summary, a qubit can be in a continuum of states between $|0\rangle$ and $|1\rangle$ until it is measured. An n-qubit quantum-mechanical system is described by the tensor product space of n two-dimensional complex vector spaces. That is, an n-qubit quantum-mechanical system has 2^n computational basis states $|x_{1x_2}...x_n\rangle$ or equivalently $|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$ where the notation $\otimes$ denotes the tensor product and $x_i \in \{0, 1\}$ for all $i \in \{0, 1, \ldots, n-1\}$. Another way to characterise a quantum-mechanical system of n qubits is with the density operator, denoted as ρ, which is a Hermitian matrix of size $2^n \times 2^n$. The density matrix can be seen as a statistical mixture of pure states of the quantum-mechanical system [2].

Given the state $|\psi\rangle$ describing a quantum mechanical system, how does it change with time? The second postulate of quantum mechanics answers this question. It states that the evolution of a closed quantum system is described by a unitary transformation. Mathematically, if the quantum state is $|\psi\ (t_1)\rangle$ at time t_1, then later t_2, the state of the quantum system is given by the following relation [2]:

$$|\psi(t_2)\ \rangle = U(t_1, t_2)|\psi(t_1)\rangle, \tag{2}$$

where $U\ (t_1,\ t_2)$ is a unitary operator that depends only on t_1 and t_2. Based on the Schrödinger's equation and the second postulate of quantum mechanics, we can obtain the following equation for the unitary $U\ (t_1,\ t_2)$:

$$U(t_1, t_2) = \exp\left(\frac{iH(t_2 - t_1)}{\hbar}\right), \tag{3}$$

where i denotes the imaginary unit and H is a fixed Hermitian operator known as the Hamiltonian (a Hermitian matrix of size $2^n \times 2^n$) of the closed quantum-mechanical system and $\hbar$ is Planck's constant. In general, for a closed quantum-mechanical system, the time evolution is given by a unitary operator $U = e^{-iHt}$ or some Hermitian operator H (where the constant $\hbar$ is absorbed in the Hamiltonian H). If the system is described in terms of the density matrix $\rho(t)$ at time t, then the time evolution for a time interval τ is given by [2]

$$\rho(t + \tau) = e^{-iH\tau}\rho(t)e^{iH\tau}. \tag{4}$$

2.2 | Qubit measurement

As discussed previously, a closed quantum-mechanical system evolves according to unitary evolution. When an external physical system interacts with this closed system to perform a measurement, the system is no longer closed. The third postulate of quantum mechanics describes the effect of a measurement on a quantum mechanical system by a set of projective operators $\{P_m\}$ such that $\sum_m P_m = I$ and $P_m P_n = \delta_{mn} P_m$ (where δ_{mn} is the Kronecker delta function). The probability to obtain the measurement outcome m for the state ρ is given by $p(m) = Tr\ [P_m \rho]$ where $Tr\ [\cdot]$ denotes the trace of a matrix. The state after the measurement is given by $P_m \rho P_m / p(m)$, that is, the projective measurement modifies the quantum-mechanical system. By repeating the projective measurements, after preparing the quantum state again, we can calculate average values $\langle O\rangle = Tr\ [O\rho]$ of an observable given by $O = \sum_m a(m) P_m$ according to its spectral decomposition, where $a(m)$, $\forall m$ are the eigenvalues of O [2].

2.3 | Quantum reservoir dynamics

The nodes of the quantum reservoir network are given by the orthogonal basis of the quantum states. This means that for n qubits, we have 2^n basis states. To be efficient, quantum reservoir computing requires complex and bounded dynamics and large network sizes [21]. It is known that these requirements can be naturally met by interacting quantum-mechanical spin systems [21]. These spin systems have (i) state space that scales exponentially with the number of spins and (ii) complex dynamics governed by unitary operators.

In this study, we consider the extensively studied model called the fully connected transverse field Ising (FC-TFI) model [10, 21]. The Hamiltonian of this model is given by

$$H = \sum_{i,j} J_{i,j} X_i X_j + h_i Z_i, \tag{5}$$

where X_i and Z_i are the Pauli X and Z operators at qubit i, the coefficient h_i denotes the coupling to an external magnetic field of qubit i and the coefficient $J_{i,j}$ denotes the inter-qubit interactions. In the FC-TFI model, all the qubits, as shown in (5), interact with each other in the x-direction and are coupled to an external magnetic field in the z-direction.

In the QRC framework, an input sequence of length T, $u\ (t) \in \{u_0, u_1, \ldots, u_{T-1}\}$ where $u_i \in [0, 1]$ is given by the trajectory of a mobile user (i) is injected into the quantum reservoir network, (ii) evolves according the quantum dynamics and then (iii) is extracted as output to be analysed. The idea of the QRC framework [10, 21] is to inject the input at time t, given by $u(t)$, through the first qubit by setting the state of this qubit to

$$|\psi_{u(t)}\rangle = \sqrt{1 - u(t)}|0\rangle + \sqrt{u(t)}|1\rangle, u(t) \in [0, 1]. \tag{6}$$

The density matrix of the system is then given by

$$\rho = \left|\psi_{u(t)}\ \rangle\ \langle\psi_{u(t)}\right| \otimes Tr_1\ [\rho], \tag{7}$$

where $Tr_1\ [\rho]$ denotes the partial trace operator over the first qubit. In other words, the quantum state of the system is given by the tensor product of the first qubit's state $|\psi_{u(t)}\rangle\langle\psi_{u(t)}|$ and the remaining qubits' states $Tr_1\ [\rho]$. After injecting the input $u\ (t)$ into the first qubit, the quantum-mechanical system continues evolving for a time τ according to the Schrödinger's equation as discussed previously in (2) and (4). The dynamics of the quantum reservoir is governed by (4) during the time τ and the information that was encoded in the first qubit will spread through the quantum reservoir. An illustrative example of the QRC is given in Figure 1. It is clear from Figure 1 that the QRC approach allows to project the input signal (the time series corresponding to the mobile user trajectory) into a high-dimensional space and, from the dynamics of the reservoir, extract outputs corresponding to the evolution of the Pauli-Z operator's average values of the selected qubits. Note that if the input signal has length T and we inject into the reservoir the values of the signal one at a time at each time step τ, then the simulation will take a total of $T\tau$ times.

For the system to evolve for a time τ, one must apply the evolution operator according to $e^{-iH\tau}\rho\ (t)e^{iH\tau}$. On a gate-based quantum computer, this can be achieved by applying the Suzuki—Trotter decomposition [2] which states that the evolution of an operator consisting in a sum of local operators for a time τ can be expressed as the evolution for a time (τ/κ) of a product of local operators, repeated κ times. In the limit of $\kappa \rightarrow \infty$, this formula is exact. More formally, if $H = \sum_i^{h_i}$, then $e^{-iH\tau} = e^{-i\sum_i^{h_i\tau}} = \lim\limits_{n\rightarrow\infty}\left(\prod\limits_i e^{-ih_i\tau/\kappa}\right)^{\kappa}$. For finite values of κ, the error made by this approximation is proportional to τ^2/κ. Since the Hamiltonian of our QRC system is based on the FC-TFI model, our task consists in implementing the evolution of the two qubits interaction operator $e^{-iJ_{i,j}X_iX_j\tau/\kappa}$ and the single qubit operator $e^{-ih_iZ_i\tau/\kappa}$. With the rotation gates defined by $R_{\{X,Z\}}(\theta) = e^{-i\frac{\theta}{2}\{X,Z\}}$, we obtain:

$$e^{-\frac{ih_iZ_i\tau}{\kappa}} = R_z\left(\frac{2h_i\tau}{\kappa}\right). \quad (8)$$

To obtain the evolution of the interaction term $e^{-iJ_{i,j}X_iX_j\tau/\kappa}$, we note that the unitary operator $e^{-i\frac{\theta}{2}ZZ}$ is realised by the sequence of gates shown in Figure 2 and that we have the identity $X = \boldsymbol{HZH}$ where $\boldsymbol{H}$ corresponds to the Hadamard gate. Thus, $e^{-iJ_{i,j}X_iX_j\tau/\kappa}$ is realised, as shown in Figure 3, with four Hadamard gates and the block given in Figure 2.

Figure 3 illustrates the basic idea of how to simulate the time evolution of the Hamiltonian given in (5) with full coupling between three qubits. Given the Hamiltonian H, which is given by $h\ (Z_0 + Z_1 + Z_2) + J_{0,1}X_0X_1 + J_{1,2}X_1X_2$ for three qubits, we can use IBM gate-based quantum computers and implement the quantum circuit given in Figure 3 to estimate the time evolution of the Hamiltonian.

The objective of the mobility trajectory prediction problem is to find, using the QRC system and given the trajectory of a mobile user $u(t)$, a non-linear function $f(u(t))$ such that the prediction error (e.g., mean-squared error) between $f(u(t))$ and a target output $\overline{u}(t)$ (teacher output) is minimised. The role of the QRC system is to emulate this non-linear function and thus produce a highly complex and non-linear system capable of predicting temporal input data. The output signal, $x_i(t)$, obtained after measuring qubit i at each time t is used at the output layer of the QRC system and combined with the readout weights $\boldsymbol{W^{out}}$ to perform the prediction. This is done using a simple linear regression model. Let $\boldsymbol{x}(t)$ denotes the vector $[x_1(t), \ldots, x_n(t)]^{\top}$. After collecting enough pairs of input-output signals $(u(t), \boldsymbol{x}(t))$ for each time t, we use a linear regression technique to train the readout weight $\boldsymbol{W^{out}}$ of size $n + 1$ where n is the number of qubits used, and the +1 is used for the bias. This can be done simply using the Moore–Penrose pseudo inverse, that is, we need to solve the following linear system:

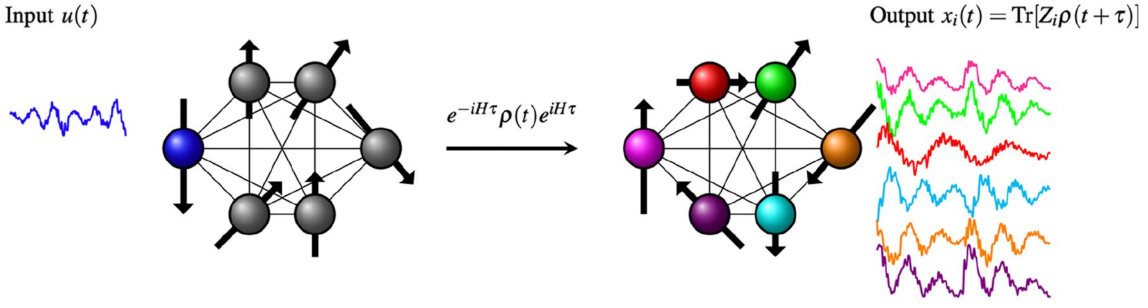
Input $u(t)$
$e^{-iH\tau}\rho(t)e^{iH\tau}$
Output $x_i(t) = \mathrm{Tr}[Z_i\rho(t+\tau)]$

FIGURE 1 An example of a QR with 6 qubits shown with grey balls (one qubit is in blue colour to indicate the input qubit). Each qubit's state is given by the corresponding arrow. An input signal (a time series) of length $\boldsymbol{T}$ is injected into one qubit at some initial time $\boldsymbol{t}$ and the system evolves for a time $\boldsymbol{\tau}$ according to the dynamics of the QR. After a time $\boldsymbol{\tau}$, a measurement is performed. By repeating this process multiple times for the same input signal, we obtain the average spin values in the z-direction of each spin (qubit). Note that if the input signal has length $\boldsymbol{T}$ and we inject into the reservoir the values of the signal one at a time at each time step $\boldsymbol{\tau}$, then, the simulation will take a total of $\boldsymbol{T\tau}$ times.

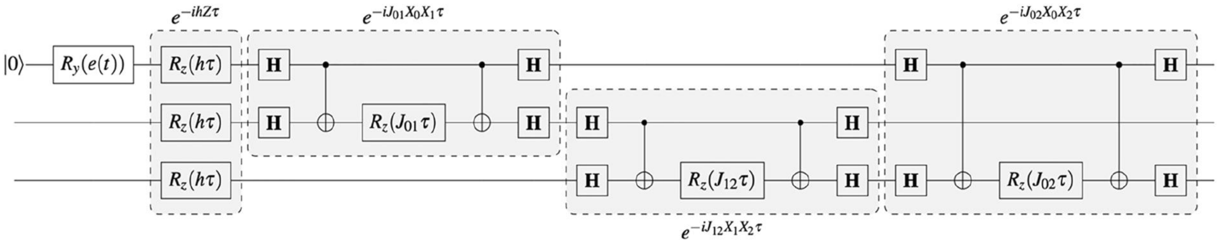

FIGURE 2 Simulation of the time evolution of the Hamiltonian $\boldsymbol{ZZ}$, that is, $\boldsymbol{e^{-iZZt}}$.

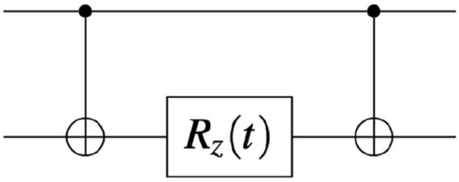

FIGURE 3 Quantum circuit realisation to calculate the exponential of the FC-TFI Hamiltonian with 3 qubits. The input is injected onto the first qubit which is prepared in the state $|0\rangle$. Then, the time series data u(t) is encoded in the first qubit using the single-qubit rotation gate about the Y-axis, $\boldsymbol{R_y}\,(\boldsymbol{e}(\boldsymbol{t}))$. Here, $\boldsymbol{e}\,(\boldsymbol{t})$ is chosen such that the state of the first qubit will be $\sqrt{\boldsymbol{1-u(t)}}\,|\boldsymbol{0}\,\rangle+\sqrt{\boldsymbol{u(t)}}|\boldsymbol{1}\rangle$ just after the gate $\boldsymbol{R_y}\,(\boldsymbol{e}(\boldsymbol{t}))$, that is, $\boldsymbol{e(t)=2\,\sin^{-1}\sqrt{u(t)}}$. We assume a full coupling between the qubits so that each qubit interacts with all other qubits. The resulting quantum circuit can be simplified but is kept as is for clarity.

$$\overline{\boldsymbol{u}}=\overline{\boldsymbol{X}}\,\boldsymbol{W}^{\boldsymbol{out}}, \tag{9}$$

where $\overline{\boldsymbol{u}}$ is the target signal of size T, the matrix $\overline{\boldsymbol{X}}$ denotes the modified output signals of the QR after adding the bias of size $T \times (n+1)$.

3 | CLASSICAL BENCHMARK APPROACHES

3.1 | LSTM prediction

To compare the QRC approach, we implemented the well-known RNN-based approach of LSTM [32]. RNN suffers from the vanishing/exploding gradient problem in which long-term gradients that are back-propagated can either vanish (reach zero) or explode (reach infinity). LSTM was proposed to deal particularly with the vanishing gradient problem. In the LSTM approach, a RNN can keep track of long-term dependencies in the input sequences thanks to the use of feedback connections and special neural network units (which allow information to persist). In LSTM, there is a key component called the cell state to which information are added or removed using other components called gates (e.g., forget gate layer, input gate layer, etc.). Other details about the proposed LSTM network are described in Section 4.

3.2 | ESN prediction

We compared the QRC approach to its classical counterpart, the echo-state network (ESN) [33]. In the ESN approach, a RNN is also used but in the context of RC. The difference between RNN and RC is that in the former the hidden layers weights are optimised during training but in the latter, these weights are randomly generated and thus are fixed during training. In ESN, only the weights of the output layers are trainable. The structure of the reservoir (i.e., the hidden layers) should be constructed to generate highly complex dynamics so that the training of the output weights reproduces with accuracy specific temporal patterns. An input signal will be injected into the reservoir that will generate a high-dimensional signal using its internal reservoir states (its dynamics). Next, the high-dimensional signal is passed to the output layer for training. Other details about the proposed ESN are also given in Section 4.

4 | EXPERIMENTAL EVALUATION

In this section, we evaluate experimentally using a real-world dataset the performance of the QRC and compare it to the LSTM and ESN approaches. Our experiments are implemented in a simulation environment as well as in small-scale quantum computers as discussed in the sequel. We use a GPS dataset called the Geolife dataset [15, 28–30] that contains GPS trajectories of **182** mobile users, that is, it contains the longitude and latitude of each mobile user. It was collected between the year **2007** and the year **2012**. Each mobile user in the Geolife dataset has its own trajectory, meaning that the length of each trajectory (the amount of data of available points) varies from one user to another. Our objective is to accurately predict future mobile user locations. We have randomly chosen three users from this dataset, which are indexed as user **#1**, user **#153** and user **#175** and we used their trajectories as movement history to learn their mobility behaviours using the QRC approach and compared to the LSTM and ESN approaches. Note that the LSTM and ESN approaches are not tuned to optimality. We only perform a grid search method to select good LSTM and ESN parameters. The aim is to illustrate how well can the QRC approach perform compared to classical approaches.

To keep the prediction procedure simple (especially for the QRC approach), we selected a part of each user's trajectory that contains approximately **200** data points. Specifically, first, for user **#1**, we selected the trajectory recorded between **24/10/2008** at **02:09:59** and **24/10/2008** at **02:47:06**, which contains **244** data points. Second, for user **#153**, we selected the trajectory recorded between **22/08/2008** at **13:10:34** and **22/08/2008** at **13:14:04**, which contains 211 data points. Finally, for user **#175**, we selected the trajectory recorded between **07/12/2007** at **23:07:44** and **08/12/2007** at **01:27:35**, which contains **194** data points. The time series data of these three users are illustrated in Figures 4 and 5 in which the former figure shows the latitude variable of the corresponding user, while the latter figure shows its longitude variable. Note that, working with very long user trajectories (thousands of data points) is also interesting but more complex especially using quantum computers. We keep the study of this interesting challenge for our future work.

Our implementation is performed in the Python programming language using a MacBook Pro with Apple M1 Pro chip, an **8**-core CPU and a **16** GB RAM. Our implementation is divided in two parts: a simulation-based part and a real-based part. For the simulation (more details are given in Subsection 4.3), we used the Qiskit library, which is an open-source software developed by IBM Research [34] to allow working with quantum computers at the level of circuits, pulses and

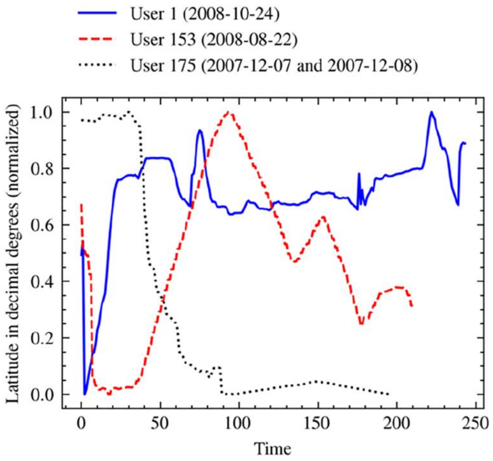

FIGURE 4 Overview of the latitude variable for the three users.

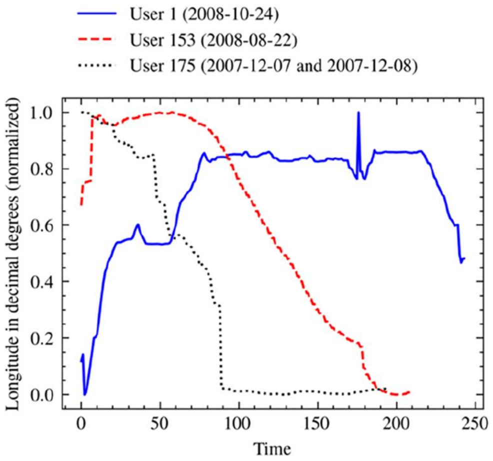

FIGURE 5 Overview of the longitude variable for the three users.

algorithms. For the real implementation, instead of accessing the Qiskit Qasm simulator, we access IBM gate-based quantum computers through the cloud quantum computing service of IBM Quantum Experience, and we execute our Python code (more details on the used quantum computers are given in Subsection 4.5).

In the sequel, first, we discuss the LSTM and the ESN architectures. Then, we discuss the QRC architecture. Finally, we present the results of the prediction.

4.1 | LSTM architecture and parameters

The Keras library in Python is used to implement the deep learning architecture of LSTM. We used a sequential architecture with three layers, two LSTM layers and one dense layer. The first LSTM layer implements **256** LSTM units with return output sequences set to true (return the last output in the output sequence). The next LSTM layer contains 128 LSTM units with return output sequences set to false (return the full sequence). The last layer is a dense layer, and it contains **2** units where the best unit is always selected for prediction. Between each layer there is a dropout of **0.2** that is added. The TensorFlow optimiser RMSprop is used for training with a learning rate of **0.0001** and the mean-squared error function as the loss function. The training lasted for a period of **2000** epochs, where a batch of size **32** is used. The prediction is made for the last **30** timesteps of each time series and the remaining $\boldsymbol{\ell - 30}$ are used for training (ℓ denotes the length of each time series). The training data are constructed using a sliding window of **70** timesteps, that is, we built a training data of **70** features.

4.2 | ESN architecture and parameters

We used the pyESN library [35] in Python. An ESN is created in pyESN with one input unit, one output unit and **500** reservoir units. The spectral radius of the recurrent weight matrix is set to **0.95** to guarantee the echo-state property. A sparsity of **0.1** is chosen, which represents the proportion of the recurrent weights set to zero. A noise of **0.001** is added to each neuron (used for regularisation). We perform a **2**-step prediction for **15** timesteps; for a total length of **30** timesteps.

4.3 | QRC architecture and parameters

We used the Python library Qiskit [34] to implement the QRC framework, and we used the Sci-Kit learn library to perform the prediction by using the linear regression. We implemented a QRC system with **4** qubits with a magnetic coupling $\boldsymbol{h = 0.5}$. The Hamiltonian with a full interaction between the qubits is assumed where every qubit interacts with all other qubits. The inter-qubit interaction coefficients $\boldsymbol{J_{i,j}}$ are chosen randomly by using a beta distribution of parameters $\boldsymbol{\alpha = \beta = 0.9}$. The time series data are injected into the QRC with a washout period of **70** timesteps to forget the dependence to the initial conditions. The QRC is left evolving according to the Hamiltonian evolution, and the gate-based quantum circuits shown in Figure 3 are implemented in IBM quantum computers using the Qiskit Qasm simulator. The quantum experiment is repeated **1024** times for each entry in the time series to obtain the average values as outputs of the QRC. A training is then performed to obtain the weight matrix $\boldsymbol{W^{out}}$ using the Moore–Penrose pseudo inverse.

4.3.1 | Notes on the user of real quantum devices

We also implemented the QRC approach in the IBM real quantum computer using five qubits. More details are given in the sequel (see Subsection 4.5). Note that using quantum computers with more qubits is a promising avenue for solving more interesting and challenging mobility management problems in ITSs that we will study in the future. Our current work will provide useful insights and important future research directions and surely will contribute to the advancement of knowledge. Further, our work can be used to evaluate other complex quantum machine learning solutions.

4.4 | Simulation results

First, we calculate, for each method, the mean squared error (MSE) between the **30**-step predicted output and the original data. The MSE of all three methods are illustrated in Tables 1–3. We can see that the QRC approach offers accurate predictions for different time series variable (longitude vs. latitude) and for different time series data (different users) even though only **4** qubits were used. On the contrary, the RNN approaches (either ESN or LSTM) used lots of neurons (**256** in LSTM and **500** in ESN) to only produce inferior results compared to the QRC approach. One of the main ideas behind using QRC was to take advantage of its capacity to produce a complex dynamical system through the Hamiltonian evolution and the high-dimensional complex Hilbert space. Tables 1–3 show that indeed quantum computing can be used to predict, with low MSE, complex time series data. Note, however, that the application of the QRC approach to more complicated time series (longer or multivariate) is important and should be analysed in our future work. It will be probably evident that for more complex time series data, the number of qubits should be increased and, thus, the trade-off between performance in terms of prediction and complexity in terms of time should be analysed carefully for the QRC approach. In the next figures, we visualise the **30**-step predictions produced by each method; QRC, ESN and LSTM.

TABLE 1 Mean squared error for user **#1**.

	Variables	
	Latitude	Longitude
QRC	**0.000131**	**0.0013**
ESN	0.00302	0.00262
LSTM	0.00453	0.00771

Note: Bold values are used to indicate the best value (the smallest error) in each column.

TABLE 2 Mean square error for user **#153**.

	Variables	
	Latitude	Longitude
QRC	**0.000213**	0.00005
ESN	0.00069	0.000236
LSTM	0.00176	0.0038

Note: Bold values are used to indicate the best value (the smallest error) in each column.

TABLE 3 Mean square error for user **#175**.

	Variables	
	Latitude	Longitude
QRC	**0.000043**	**0.000022**
ESN	0.000376	0.000276
LSTM	0.000166	0.00004

Note: Bold values are used to indicate the best value (the smallest error) in each column.

Figure 6 illustrates the prediction produced by the QRC approach on the time series data of the three users for the latitude variable. We can see that the prediction is very close to the true trajectory data.

Figure 7 shows a visualisation of the prediction produced by the QRC approach on the time series data of the three users for the longitude variable. The QRC prediction is again very accurate as the predicted signal is very close to that of the true data. Despite the change of the time series variable, the QRC can produce a good prediction for all users, which illustrate the remarkable feature of the QRC system. When the time series is like a straight line (the prediction for user **#175** in Figure 7), the prediction might not look smooth and perfect, but the

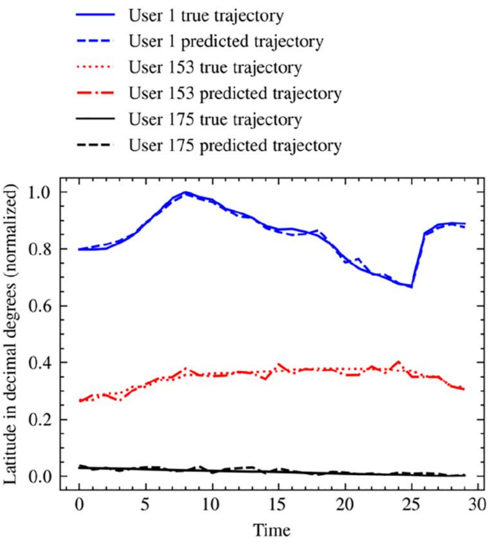

FIGURE 6 Quantum reservoir computing (QRC) **30**-steps prediction of the latitude variable for the three users.

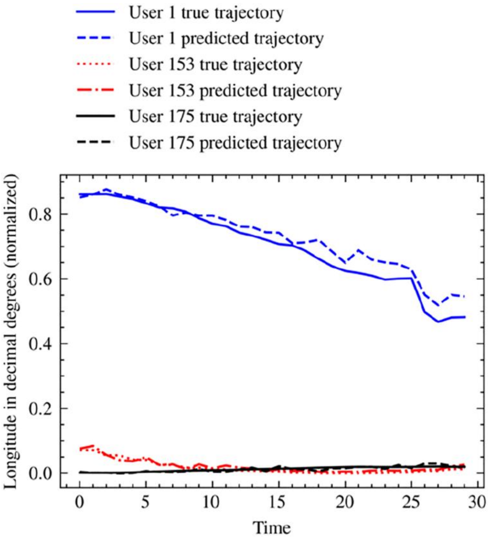

FIGURE 7 QRC **30**-steps prediction of the longitude variable for the three users.

MSE is still very low. For this user, its original time series is almost constant especially for the last **100** timesteps, which is caused by an overfitting problem.

In Figures 8 and 9 we visualise the prediction produced by the ESN approach on the time series data of the three users for the latitude variable and the longitude variable, respectively. The ESN approach produces also good prediction as the predicted signals, and their corresponding true values are close to each other. Further, the ESN approach produces similar predictions compared to the QRC approach. Nonetheless, the latter is better and has lower MSE as well. It is also true for the ESN approach, as in the QRC approach, that the prediction for the user **#175** is more difficult than the remaining users.

In Figures 10 and 11 we visualise the prediction produced by the LSTM approach on the time series data of the three users for the latitude variable and the longitude variable respectively. The LSTM prediction is the worst among all approaches. This was also observed in Tables 1–3, where the MSE was the worst for the LSTM prediction for all users (except for user **#175**).

4.5 | Real quantum implementation results

In this part, we obtain the QRC prediction of the longitude time series variable for user **#1** with a real quantum computer, and we compared the result to the ESN and LSTM predictions. We access IBM quantum computers through the provider ibm-q/open/main/that provides access to six real quantum computers. We implement the QRC approach on the version 1.1.34

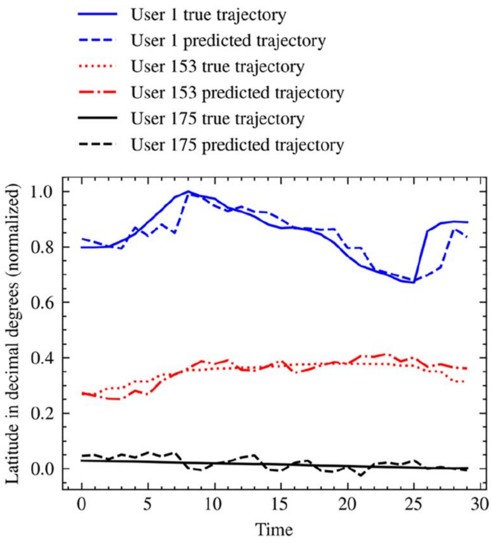

FIGURE 8 Echo-state network **(ESN) 30**-steps prediction of the latitude variable for the three users.

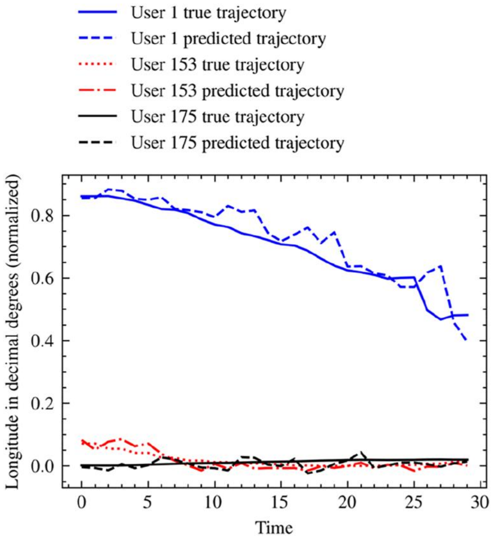

FIGURE 9 Echo-state network (ESN) **30**-steps prediction of the longitude variable for the three users.

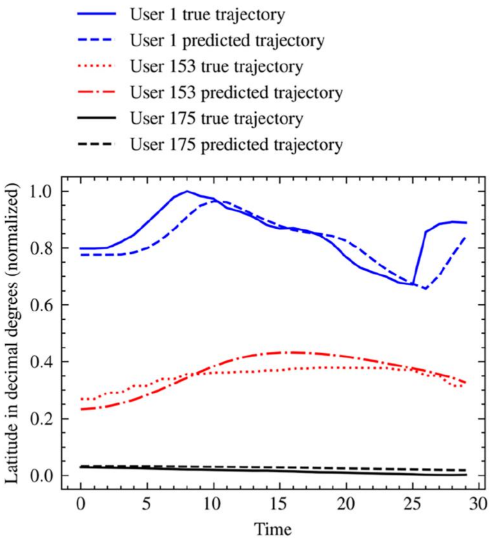

FIGURE 10 Long short-term memory (LSTM) **30**-steps prediction of the latitude variable for the three users.

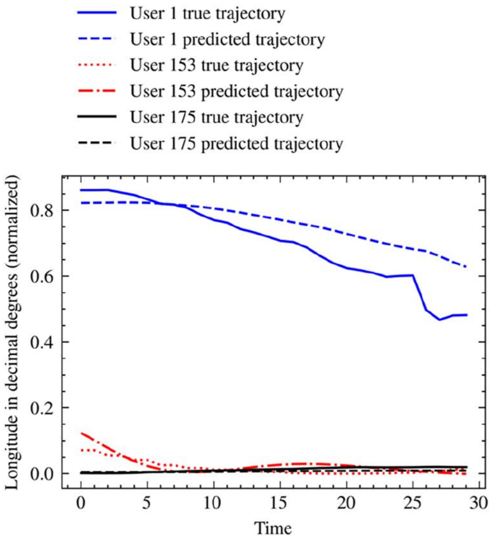

FIGURE 11 Long short-term memory (LSTM) **30**-steps prediction of the longitude variable for the three users.

of the ibmq-quito **5**-qubits quantum computer that is equipped with a processor of type Falcon r4T. Due to limited available resources on IBM quantum computers, that is, only a maximum of **20,000** shots and a maximum of **100** circuits are permitted, we selected a small-size time series to implement the QRC approach. We selected the longitude of the user **#1** recorded between **2008, 10−27** at **11:54:49** and **2008, 10−27** at **12:05:54**, which contains 50 data points. We used 10 timesteps for the washout period, and we performed a **5**-timesteps prediction. Due to the natural quantum noise and gate errors, we increased the number of shots to **4000** and we increased the optimisation level to **3** to optimise the quantum circuits at the expense of longer transpilation time.

We observe from Figure 12 that the LSTM prediction produces very close performance to the QRC prediction, but the latter is still better. Even though current real quantum computers are not fault-tolerant, we demonstrated that the QRC prediction approach is able to produce satisfactory results in terms of MSE compared to the well-known and most used deep learning-based prediction approaches such as the LSTM and the ESN approaches. Note that the prediction is done on small-size time series and a more sophisticated quantum prediction approach might be needed for more complex time series. Despite this, the QRC approach is appealing and worth considering in real-world scenarios.

4.6 | Discussions

We observed through our analysis of the different results that implementing the QRC approach on open-access IBM quantum experience is easy, especially for small-size time series and when the number of qubits is not large. However, for large-size time series or when the number of qubits is large, one needs more resources in terms of time and computation. We observed that more advanced quantum computers can forecast efficiently more challenging and interesting time series machine learning problems compared to their classical counterparts. Also, the QRC approach is appealing to solve wireless networks machine learning tasks such as the prediction and classification of time series, for example, the prediction of wireless traffic, the prediction of signal-related measurements etc.

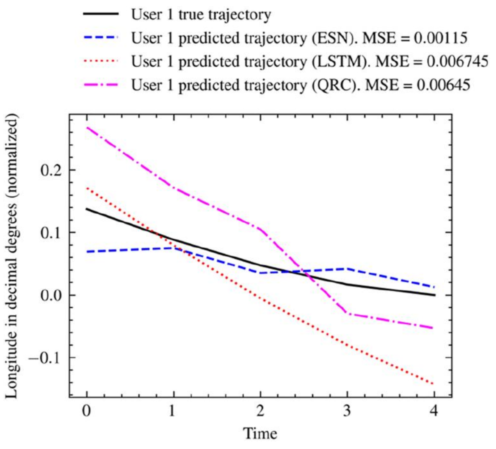

FIGURE 12 Real quantum computer **5**-steps prediction of the longitude variable for user **#1**.

4.7 | Future works

In the future, it is worth investigating the QRC approach for multivariate time series that have much longer length. It would also be interesting to investigate the trade-off between complexity and performance of the QRC when using a higher number of qubits. Another research axis that we intend to pursue is hyper-parameters optimisation. In the case of the QRC, inter-qubit interactions and evolution time should be tuned to measure the accuracy of the trajectory prediction for different memory capacities of the reservoir. To establish a fair comparison between the quantum and the classical approaches, the hyper-parameters of the classical models should also be optimised. Finally, the characterisation of the impact of the quantum device's noise for time series of different length and washout length will be the object of future research.

5 | CONCLUSIONS

In this study, we applied a quantum reservoir computing (QRC) approach, proposed recently to predict time series data, to the user mobility prediction problem in mobile wireless networks. We used a real-world GPS dataset with hundred timesteps. The QRC approach exploited the rich dynamics of quantum mechanics to produce a highly complex and dynamical quantum system capable of highly producing accurate prediction of GPS trajectories. The QRC approach was compared to two classical approaches; the long-short term memory (LSTM) and the echo-state network (ESN). Both classical approaches are based on the recurrent neural network framework. The ESN uses the reservoir computing technique like the quantum one. The difference is that the quantum technique constructs a quantum reservoir based on the quantum mechanics, whereas the ESN constructs a classical reservoir based on random matrix generation. The quantum reservoir used the time evolution of the fully connected transverse field Ising Hamiltonian, and we showed how we can produce this time evolution using quantum simulation with gate-based quantum circuits. We showed that the QRC approach can produce the lowest prediction error in terms of mean-squared error compared to the LSTM and the ESN approaches.

AUTHOR CONTRIBUTIONS

Zoubeir Mlika and Soumaya Cherkaoui defined the research objectives of the article. All authors participated to the development of the methodology. Zoubeir Mlika and Jean Frédéric Laprade implemented the algorithm, Zoubeir Mlika performed the simulations. Zoubeir Mlika and Soumaya Cherkaoui conducted the analysis. All authors contributed to write this manuscript.

ACKNOWLEDGEMENTS

The authors would like to thank the Fonds de recherche du Québec - Nature et technologies (FRQNT) as well as the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial supports.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in (Geolife GPS trajectory dataset) at https://doi.org/10.1145/1409635.1409677, reference number [28].

ORCID

Zoubeir Mlika https://orcid.org/0000-0003-3417-7704

REFERENCES

1. Feynman, R., Leighton, R., Sands, M.: The Feynman Lectures on Physics, Vol. III: The New Millennium Edition: Quantum Mechanics. The Feynman Lectures on Physics (2011)
2. Chuang, a.M.A.N.I.L., Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2011)
3. Hassija, V., et al.: Present landscape of quantum computing. IET Quant. Commun. 1(2), 42–48 (2020). https://doi.org/10.1049/iet-qtc.2020.0027
4. Hidary, J.: Quantum Computing: An Applied Approach. Springer International Publishing (2019)
5. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Annual Symposium on Foundations of Computer Science (1994)
6. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304(5667), 78–80 (2004). https://doi.org/10.1126/science.1091277
7. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. 14(11), 2531–2560 (2002). https://doi.org/10.1162/089976602760407955
8. Verstraeten, D., et al.: An experimental unification of reservoir computing methods. Neural Network. 20(3), 391–403 (2007). https://doi.org/10.1016/j.neunet.2007.04.003
9. Dambre, J., et al.: Information processing capacity of dynamical systems. Sci. Rep. 2(1), 1–7 (2012). https://doi.org/10.1038/srep00514
10. Fujii, K., Nakajima, K.: Harnessing disordered-ensemble quantum dynamics for machine learning. Phys. Rev. Appl. 8(2), 024030 (2017). https://doi.org/10.1103/physrevapplied.8.024030
11. Kim, H., Ben-Othman, J.: A successive epidemic prevention infrastructure using mobile robots and smart devices in intelligent public area. IEEE Commun. Lett. 26(3), 692–695 (2022). https://doi.org/10.1109/lcomm.2021.3138761
12. Kim, H., et al.: Intelligent aerial-ground surveillance and epidemic prevention with discriminative public and private services. IEEE Netw. 36(3), 40–46 (2022). https://doi.org/10.1109/mnet.002.2100510
13. Li, J., et al.: Deep reinforcement learning-based mobility-aware robust proactive resource allocation in heterogeneous networks. IEEE Trans. Cogn. Commun. Netw. 6(1), 408–421 (2019). https://doi.org/10.1109/tccn.2019.2954396
14. Filali, A., et al.: Preemptive SDN load balancing with machine learning for delay sensitive applications. IEEE Trans. Veh. Technol. 69(12), 15947–15963 (2020). https://doi.org/10.1109/tvt.2020.3038918
15. Bahra, N., Pierre, S.: RNN-based user trajectory prediction using a preprocessed dataset. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Thessaloniki (2020)
16. Bahra, N., Pierre, S.: A bidirectional trajectory prediction model for users in mobile networks. IEEE Access 10, 1921–1935 (2021). https://doi.org/10.1109/access.2021.3139867
17. Xu, C., et al.: Offloading elastic transfers to opportunistic vehicular networks based on imperfect trajectory prediction. IEEE/ACM Trans. Netw. 31(1), 279–293 (2022). https://doi.org/10.1109/tnet.2022.3189047
18. Dinani, M.A., et al.: Gossip learning of personalized models for vehicle trajectory prediction. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW). Nanjing, China (2021)
19. Ip, A., Irio, L., Oliveira, R.: Vehicle trajectory prediction based on LSTM recurrent neural networks. In: IEEE Vehicular Technology Conference (VTC2021-Spring). Helsinki (2021)
20. Liu, X., et al.: Trajectory design and power control for multi-UAV assisted wireless networks: a machine learning approach. IEEE Trans. Veh. Technol. 68(8), 7957–7969 (2019). https://doi.org/10.1109/tvt.2019.2920284
21. Kutvonen, A., Fujii, K., Sagawa, T.: Optimizing a quantum reservoir computer for time series prediction. Sci. Rep. 10(1), 1–7 (2020). https://doi.org/10.1038/s41598-020-71673-9
22. Feng, Z.-K., et al.: Monthly runoff time series prediction by variational mode decomposition and support vector machine based on quantum-behaved particle swarm optimization. J. Hydrol. 583, 124627 (2020). https://doi.org/10.1016/j.jhydrol.2020.124627
23. Araujo, R.D.A., de Oliveira, A.L., Soares, S.C.: Aquantum-inspired hybrid methodology for financial time series prediction. In: IEEE International Joint Conference on Neural Networks (IJCNN). Barcelona (2010)
24. Ahmed, A.S.A., Kurnaz, S.: Quantum computing and artificial neural network for classification and time series prediction. In: IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). Istanbul, Turkey (2022)
25. Suzuki, Y., et al.: Natural quantum reservoir computing for temporal information processing. Sci. Rep. 12(1), 1–15 (2022). https://doi.org/10.1038/s41598-022-05061-w
26. Hellstern, G.: Analysis of a hybrid quantum network for classification tasks. IET Quant. Commun. 2(4), 153–159 (2021). https://doi.org/10.1049/qtc2.12017
27. Dutta, S., et al.: Quantum circuit design methodology for multiple linear regression. IET Quant. Commun. 1(2), 55–61 (2020). https://doi.org/10.1049/iet-qtc.2020.0013
28. Zheng, Y., et al.: Understanding transportation modes based on GPS data for web applications. ACM Trans. Web 4(1), 1–36 (2010). https://doi.org/10.1145/1658373.1658374
29. Zheng, Y., et al.: Understanding mobility based on GPS data. In: ACM Conference on Ubiquitous Computing (UbiComp). Seoul (2008)
30. Zheng, Y., et al.: Learning transportation modes from raw GPS data for geographic application on the web. In: ACM International Conference on World Wild Web (WWW). Beijing (2008)
31. Codeca, L., Frank, R., Engel, T.: Luxembourg sumo traffic (LUST) scenario: 24 hours of mobility for vehicular networking research. In: IEEE Veh. Netw. Conference (VNC). Kyoto (2015)
32. Yu, Y., et al.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31(7), 1235–1270 (2019). https://doi.org/10.1162/neco_a_01199
33. Bianchi, M., et al.: Reservoir computing approaches for representation and classification of multivariate time series. IEEE Trans. Neural Netw. Learn. 32(5), 2169–2179 (2021). https://doi.org/10.1109/tnnls.2020.3001377
34. IBM.: Qiskit: An Open-Source Framework for Quantum Computing. [Online] (2021). https://qiskit.org. Accessed 14 2022
35. Korndörfer, C.: Echo-State Networks in Python. [Online] (2018). https://github.com/cknd/pyESN. Accessed 14 2022

How to cite this article: Mlika, Z., et al.: User trajectory prediction in mobile wireless networks using quantum reservoir computing. IET Quant. Comm. 4(3), 125–135 (2023). https://doi.org/10.1049/qtc2.12061